IMAGE STABILISATION BASED ON MOSAICS

Xevi Cufí, Rafael García and Joan Batlle

University of Girona, Computer Vision and Robotics Group,
Institute of Informatics and Automation,
17003 Girona, Spain.

email: {xcuf,rafa,jbatlle}@eia.udg.es

Abstract

We describe an image stabilization system that compensates the motion of a moving platform carrying a camera. The dynamic construction of an image mosaic allows the selection of a reference image (normally the first of the sequence) and the registration of every incoming image with the reference frame. A feature-based mosaicking system is proposed in this paper to achieve image stabilization. The creation of the mosaic is accomplished in several stages: point selection, feature matching, detection of noisy points and homography computation. Finally, as the mosaic is constructed, a virtual sequence aligned with the reference image is generated. In this work we demonstrate that the adequate use of textures as discriminative properties of the image can improve, to a large extent, the accuracy of the constructed mosaic.

1 Introduction

A mosaic is a large composite image obtained from the alignment and merging of several images showing a different view of the same scene. Image mosaics can be used for many different applications, such as image stabilization (Hansen et al. 1994), construction of visual maps for robots navigation (Xu and Negahdaripour 1997), recovery of camera and object motions, map construction from aerial or satellite photographs, video compression, etc. Two main approaches have been exploited in the literature to construct image mosaics: feature-less and feature-based methods. In both cases the aim is to find a set of 2D planar transformations *registering* every input image with the coordinates system of the mosaic. This transformation, also known as homography (Szeliski 1994), can be expressed as a matrix multiplication in homogeneous coordinates:

$$\tilde{\mathbf{x}}_i = \mathbf{H} \cdot \tilde{\mathbf{x}}_i^{(k)} \quad \text{or} \quad \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_i^{(k)} \\ y_i^{(k)} \\ 1 \end{bmatrix} \tag{1}$$

where $\tilde{\mathbf{x}}^{(k)} = (\mathrm{x}_1^{(k)}, \mathrm{x}_2^{(k)}, \mathrm{x}_3^{(k)})$ are the homogeneous coordinates of a 2D point $\mathbf{x}^{(k)}$ defined in the present image (*I*), being $\mathbf{x}^{(k)} = (u^{(k)}, v^{(k)}) = (\mathrm{x}_1^{(k)} / \mathrm{x}_3^{(k)}, \mathrm{x}_2^{(k)} / \mathrm{x}_3^{(k)})$ its corresponding Cartesian coordinates; and $\tilde{\mathbf{x}} = (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)$ are the homogeneous coordinates of the

same point defined in the mosaic reference frame. The symbol $\cong$ indicates equality up to scale, and $h_{11}, h_{12}, ..., h_{32}$ are 8 parameters that determine the 2D projective transform. The image *registration* process consists in finding this 8-parameter set. In the case of the *feature-less* methods, it is achieved through an iterative process by minimizing the sum of the squared intensity errors over all corresponding pairs of pixels which are present in both images (Szeliski and Kang 1995). A commonly used method to solve this iterative non-linear minimisation is the Levenberg-Marquardt algorithm (Press et al. 1992). Although feature-less methods are quite accurate, they suffer from the slowness of computation, need good initial values for a good convergence, and can get stuck at local minima.

For all the reasons mentioned above we have chosen a *feature-based* approach. Equation 1 is computed from the set of features detected in the image $I^{(k)}$ and their matchings in the reference image. If more than 4 feature/matching pairs are available the system can be solved by a least squares technique. Once the best transformation **H** has been found, the acquired image can be warped into the reference frame of the mosaic. The stabilized image can be retrieved from the mosaic since it has been aligned with the previous images.

2 Motion estimation

The motion estimation phase consists on finding the parameters that describe the relationship between the present image and the reference image. It is accomplished in four steps: feature detection in the present image, feature matching in the reference image, elimination of noisy data and homography computation.

2.1 Feature detection

The searching for feature correspondences is performed in a two-step approach. First, the zones of the image presenting high spatial gradient information are selected by means of a corner detector (Harris and Stephens 1988). The idea is the following: the image is divided in small windows W_i, centered at the point $\mathbf{p}_i$ (for which the motion is to be estimated). Then, matrix **G** is computed as follows:

$$\mathbf{G} = \sum_{W_i} \begin{pmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_u^2 \end{pmatrix}, \text{ with } I_u = \left(\frac{\partial I}{\partial x} \right) \text{ and } I_v = \left(\frac{\partial I}{\partial y} \right) \quad (2)$$

A feature is a good candidate to track if **G** is well-conditioned, that is, if both eigenvalues of **G** are above a user-defined threshold. This means that the image point $\mathbf{p}_i$ presents a rapid intensity variation on neighboring pixels in the *x* and *y* directions.

Then, the textural parameters of these areas of the image are used as a matching vector to be correlated with the next image in the sequence. Textures significantly help in the location of features in the image. The set of textural features used in our implementation has been chosen for its suitability in underwater imaging.

2.2 Feature matching

Once the corners of image $I^{(k)}$ have been obtained, the algorithm searches for the candidate matches in the reference image *I*. The matching process is accomplished in the

following way (see figure 1): For every point $\mathbf{m}_j^{(k)}$ in image $I^{(k)}$ a correlation is performed by convolving a small window centered at $\mathbf{m}_j^{(k)}$ over a search window of the reference image $I^{(k+1)}$. Then, given a corner point $\mathbf{m}_j^{(k)}$ in image $I^{(k)}$, a search for the best matches $\{\mathbf{m}_{j1}, \mathbf{m}_{j2}, ..., \mathbf{m}_{jq}\}$ is performed in the reference image. Only those matches that are quite similar to the original correlation window of $\mathbf{m}_j^{(k)}$ are taken into account. This similarity measure is computed by means of the correlation score described in (Zhang et al. 1994). The threshold of the correlation score to be considered as a candidate match has been fixed to 0.75. Once the set of possible matches $\{\mathbf{m}_{j1}, \mathbf{m}_{j2}, ..., \mathbf{m}_{jq}\}$ has been obtained, the texture parameters of the patches centered at every matching point are computed (correlation windows on the right in figure 1).

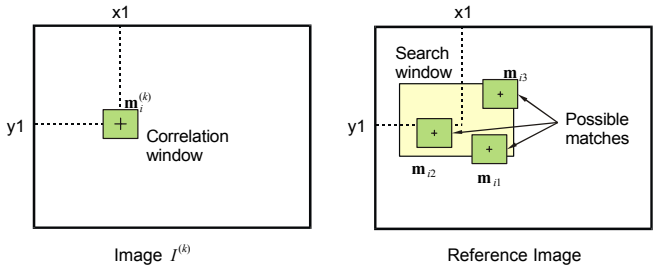


Figure 1: Typical situation where a corner point $\mathbf{m}_j^{(k)}$ has several possible matches in the reference image

For every possible matching in the reference image *I*, a vector of texture parameters is computed in the neighborhood of $\mathbf{m}_{jx}$. Three texture parameters that have been used:

(1) The *energy filters* are derived from the computation of a series of statistical measures on a pre-filtered image (Laws 1980). This component is obtained by applying a set of masks (3×3 or 5×5) that define some textural properties of the image, such as *level*, *edge*, *spot*, *wave*, *ripple* and *oscillation* (figure 2 shows the resulting masks for a 3×3 neighborhood). In order to obtain these masks, a series of vectors defining these textural proprieties are combined. Further details can be found in (Laws 1980).

```
 1  2  1        -1  0  1        -1  2 -1
 2  4  2        -2  0  2        -2  4 -2
 1  2  1        -1  0  1        -1  2 -1
  L3L3            L3E3            L3S3

-1 -2 -1         1  0 -1         1 -2  1
 0  0  0         0  0  0         0  0  0
 1  2  1        -1  0  1        -1  2 -1
  E3L3            E3E3            E3S3

-1 -2 -1         1  0 -1         1 -2  1
 2  4  2        -2  0  2        -2  4 -2
-1 -2 -1         1  0 -1         1 -2  1
  S3L3            S3E3            S3S3
```

Figure 2: 3×3 masks applied by the energy filter

(2) A second texture operator based on the spatial distribution of pixels in the image has been used: *Co-occurrence matrix* (Haralick et al. 1973). It takes into account the frequency of appearance of the pairs of pixels located at a distance d and an angle θ (co-occurrences). The set of statistics illustrated in figure (3) is computed for every co-occurrence matrix, obtaining the textural characteristics of the image.

(3) Finally, since a textured region can be described by means of its texture spectrum –that is, a set of values called texture units– a set of 3×3 simple local patterns can be defined. The different texture units can be determined from these patterns, obtaining a texture measure of the considered region. This last texture operator, known as *Local Binary Pattern* (Ojala et al. 1996), has also been used in our study.

We should take into account that the first two operators can generate several measurements, depending on the number of orientation angles, the distance of correlation and the size of the neighborhood. In our application we chose 4 different angles for the coocurrence matrix, taking only distances of 1-pixel, and image gray-levels are sub-sampled to 6 bits.
Once all the texture measures have been performed, every matching point $\mathbf{m}_j$ stores its texture characterization in a vector. This texture vector is mapped onto an N-dimensional space, where it is compared with the texture vector of the original point $\mathbf{m}_j^{(k)}$. The Euclidean distance is then computed, obtaining a texture similarity measure. After this process, a set of correspondences in image $I^{(k+1)}$ is obtained from every corner in image $I^{(k)}$, and every correspondence has two measures of similarity: correlation and texture. By averaging these two values, the reliability (r) of every match is obtained. By averaging the reliability value with the correlation score, the best candidate match is selected.

$$Probability = \max_{i,j=0}^{N-1} (m_{ij}) \qquad Entropy = \sum_{i,j=0}^{N-1} m_{ij} \times \log(m_{ij})$$

$$Uniformity = \sum_{i,j=0}^{N-1} m_{ij}^2 \qquad Contrast = \sum_{i,j=0}^{N-1} (i-j)^2 \times m_{ij}$$

$$Inverse = \sum_{i,j=0}^{N-1} \frac{m_{ij}}{(i-j)^2} \quad i \neq j \qquad Correlation = \sum_{i,j=0}^{N-1} \frac{(i-\mu)\times(j-\mu)\times m_{ij}}{\sigma^2}$$

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{m_{ij}}{1+|i-j|} \qquad \mu = \sum_{i,j=0}^{N-1} i \times m_{ij}$$

$$\sigma = \sqrt{\sum_{i,j=0}^{N-1} (i-\mu)^2 \times m_{ij}}$$

Figure 3: Statistical measures performed to characterize the texture

2.3 Eliminating outliers

After the correspondences have been found, a set of displacement vectors relating the features of two images of the sequence is obtained. Every vector relates the coordinates of the same texture-feature in both images.

Although a lot of effort is devoted to the matching procedure, some false matches (known as *outliers*) could still appear among the right correspondences, mainly due to the presence of moving objects (algae or fishes) that violate the assumption of static scene, or even to the inherent system noise. For this reason, a robust estimation method has to be applied. The Least Median of Squares (LMedS) algorithm has been implemented to reduce the distance of every matching point to its epipolar line by robustly estimating the fundamental matrix **F**. A brief description of the LMedS algorithm is given below, but a more detailed description of this method can be found in (Rousseeuw and Leroy 1987).

The principle of our implementation of the LMedS is the following: given a regression problem where *n* is the minimum number of data points which determine a solution, compute a candidate solution based on a randomly chosen *n*-tuple from the data. Then, estimate the fit of this solution to all the data, defined as the median of the squared residuals. Our regression problem is the computation of the fundamental matrix **F**, by means of the 8-point algorithm described in (Faugeras 1993). The median of the squared residuals M_{err} is defined by:

$$M_{err} = \underset{j}{med}\left(d^2\left(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{F}\tilde{\mathbf{m}}_j\right)\right) + \left(d^2\left(\tilde{\mathbf{m}}_j, \mathbf{F}^T\tilde{\mathbf{m}}_j^{(k)}\right)\right) \tag{3}$$

where $\tilde{\mathbf{m}} = (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)$ are the homogeneous coordinates of a 2D point $\mathbf{m}$ defined in the image plane, being $\mathbf{m}$ its corresponding Cartesian coordinates; and $d^2\left(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{F}\tilde{\mathbf{m}}_j\right)$ is the square distance from a point $\tilde{\mathbf{m}}_j^{(k)}$ to its epipolar line $l_j = \mathbf{F}\tilde{\mathbf{m}}_j$. It should be noted that when the point features are close to a plane, best results can be obtained if the homography matrix **H** is used instead of the fundamental matrix **F**.

2.4 Homography computation

Once a robust list of correspondences is available, the first image in the sequence is selected as a reference frame. The mosaic coordinate system is placed at the origin of this reference frame. When a new image has to be added to the mosaic, an affine transformation matrix provides its best fitting with respect to the reference image (that at the beginning of the process). The projective transform depicting the inter-frame motion is computed by finding the parameters $h_{11}, h_{12}, ..., h_{32}$ of equation 1. However we have chosen an affine model, that differs from the projective on of equation 1 in the absence of perspective deformation.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i^{(k)} \\ y_i^{(k)} \\ 1 \end{bmatrix} \tag{4}$$

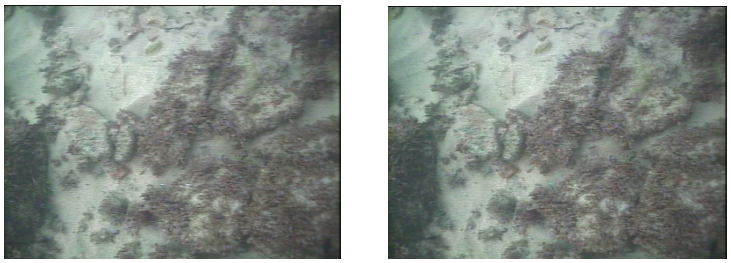
where $(x_i^{(k)}, y_i^{(k)}, 1)$ and $(x_i, y_i, 1)$ denote a correspondence point in the present $I^{(k)}$ and reference image *I*, respectively, expressed in homogeneous coordinates. $h_{11}, h_{12}, ..., h_{23}$ are the 6 parameters that determine an affine transform; and $\cong$ indicates equality up to scale. Each point correspondence generates two equations, then $n \geq 3$ points generate $2n$ linear equations that are sufficient to solve for the **H** matrix. The image registration process consists in finding this 6-parameter set, which is achieved through a least squares iterative process.

3 Registration to the mosaic

As soon as the best transformation **H** between two frames has been found, the present image can be warped with the mosaic. The 2D motion of the camera is known in pixels from one image to the next one, as an affine measure: rotation, translation and scaling. Therefore, the center of the reference frame can be taken as a fixed point, and every image can be adjusted so that this point remains motionless for the whole sequence.

4 Experimental Results

In order to demonstrate the effectiveness of the image stabilization algorithm, a small subset of images from a video sequence has been provided. The set of stabilized images has been taken from an underwater sequence, and the registration to the first image of the sequence is used to keep the position of an underwater vehicle (station keeping). Figure 4a is taken as the reference frame. The motion detected between the reference image and figure 4b is illustrated in figure 5. The resulting mosaic is shown in figure 6.



(a) (b)

Figure 4: Sample images presenting a rotation and scale difference

References

Faugeras, O. (1993) "*Three-dimensional computer vision: a geometric viewpoint*", MIT Press.

Hansen, M., Anandan, P., Dana, K., Wal, G. & Burt, P. (1994) "*Real-time scene stabilization and mosaic construction*", in Proceedings of the IEEE Workshop on Applications of Computer Vision, pp. 54-62.

Haralick, R.M. Shanguman, K. & Dinstein, I. (1973) "*Textural Features for image classification*", IEEE Transactions on Systems, Man and Cybernetics, vol. 3, pp. 610-621.

Harris, C.G. & Stephens, M.J. (1988) "*A combined corner and edge detector*", Proceedings of the Fourth Alvey Vision Conference, Manchester, pp. 147-151.

Laws, K.I. (1980) "*Textured Image Segmentation*", Ph.D. Thesis, Processing Institute, University of Southern California, Los Angeles.

Ojala, T., Pietikainen, M. & Harwood, D. (1996) "*A comparative Study of Texture Measures with Classification Based on Feature Distribution*", Pattern Recognition, vol. 29, pp. 51-59.

Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vetterling, W. T. (1992) "*Numerical Recipes in C: the art of scientific computing*", Cambridge University Press, Second Edition.

Rousseeuw P. & Leroy, A. (1987) "*Robust Regression and Outlier Detection*", John Wiley & Sons, New York.
Szeliski, R. & Kang, S.B. (1995) "*Direct methods for visual scene reconstruction*", in Proceedings of the IEEE Workshop on Representations of Visual Scenes, pp. 26-33, Cambridge, Massachusetts.
Szeliski, R. (1994) "*Image mosaicing for tele-reality applications*", in Proceedings of the IEEE Workshop on Applications of Computer Vision, pp. 44–53.
Xu, X. & Negahdaripour, S. (1997) "*Vision-based motion sensing from underwater navigation and mosaicing of ocean floor images*", in Proceedings of the MTS/IEEE OCEANS Conference, vol. 2, pp. 1412–1417.
Zhang, Z., Deriche, R., Faugeras, O. & Luong, Q.T. (1994) "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", INRIA RR-2273.

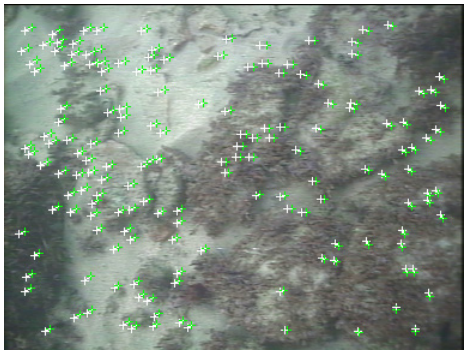


Figure 5: Selected points to compute image registration (after outlier rejection)

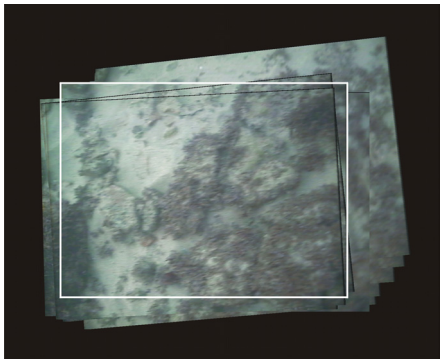


Figure 6: Stabilized sequence construction through a visual mosaic. The reference image location is enhanced with a white box.